In [40]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [41]:
```
def read_data(filename):
    return pd.read_csv(filename)
```

In [42]:
```
data=read_data("pima-indians-diabetes.csv")
data.head(5)
```

Out[42]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

In [43]:
```
def clean_data(data):
    data.isnull().sum()
    data.dropna(inplace=True)
    return data
```

In [44]:
```
data=clean_data(data)
data.head(5)
```

Out[44]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

In [45]:
```
def data_scaling(data,li):
    from sklearn import preprocessing
    scaler = preprocessing.StandardScaler()
    standard_df = scaler.fit_transform(data)
    standard_df = pd.DataFrame(standard_df, columns =li)
    return standard_df
```

In [46]:
```
df=data_scaling(data.iloc[:,:-1],["Pregnancies","Glucose","BloodPressure","SkinThickne
df.head(2)
```

Out[46]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctic
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	0.4684
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	-0.3650

In [48]:
```
data.head(3)
```

Out[48]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32

In [49]:
```
def get_train_test(df,data):
    x=df.iloc[:,:]
    y=data.Outcome
    from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20)
    return x_train,x_test,y_train,y_test
```

In [50]:
```
x_train,x_test,y_train,y_test=get_train_test(df,data)
x_train.head(2)
```

Out[50]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
30	0.342981	-0.372265	0.304734	0.342717	-0.692891	0.508619	0.22
271	-0.547919	-0.403562	-0.367337	0.719086	-0.206648	-0.862109	-1.03

In [52]:
```
y_train[:5]
```

Out[52]:
```
30     0
271    0
170    1
543    0
679    0
Name: Outcome, dtype: int64
```

In [53]:
```
x_train.shape
```

Out[53]:
```
(614, 8)
```

In [54]:
```
x_test.shape
```

Out[54]:
```
(154, 8)
```

In [ ]:
```

```

In [72]:
```
def return_model(x_train,y_train):  # hyperparamter tuning
    d={}
    from sklearn.ensemble import RandomForestClassifier
    for i in range(10,101,10):
        model = RandomForestClassifier(n_estimators =i)
        model.fit(x_train,y_train)
        d[str(i)]=model
    return d
```

In [73]:
```
d=return_model(x_train,y_train)
d
```

Out[73]:
```
{'10': RandomForestClassifier(n_estimators=10),
 '20': RandomForestClassifier(n_estimators=20),
 '30': RandomForestClassifier(n_estimators=30),
 '40': RandomForestClassifier(n_estimators=40),
 '50': RandomForestClassifier(n_estimators=50),
 '60': RandomForestClassifier(n_estimators=60),
 '70': RandomForestClassifier(n_estimators=70),
 '80': RandomForestClassifier(n_estimators=80),
 '90': RandomForestClassifier(n_estimators=90),
 '100': RandomForestClassifier()}
```

In [77]:
```
def test_all_model(d,x_test,y_test):
    out={}
    for k,v in d.items():
        out[k]=v.score(x_test,y_test)
    return out
```

In [78]:
```
out=test_all_model(d,x_test,y_test)
out
```

Out[78]:
```
{'10': 0.7402597402597403,
 '20': 0.7727272727272727,
 '30': 0.7792207792207793,
 '40': 0.8116883116883117,
 '50': 0.7727272727272727,
 '60': 0.7857142857142857,
 '70': 0.7792207792207793,
 '80': 0.8181818181818182,
 '90': 0.7922077922077922,
 '100': 0.7922077922077922}
```

In [85]:
```
score_df=pd.DataFrame(pd.Series(out),columns=["score"])
score_df
```

Out[85]:

	score
10	0.740260
20	0.772727
30	0.779221
40	0.811688
50	0.772727
60	0.785714
70	0.779221
80	0.818182
90	0.792208
100	0.792208

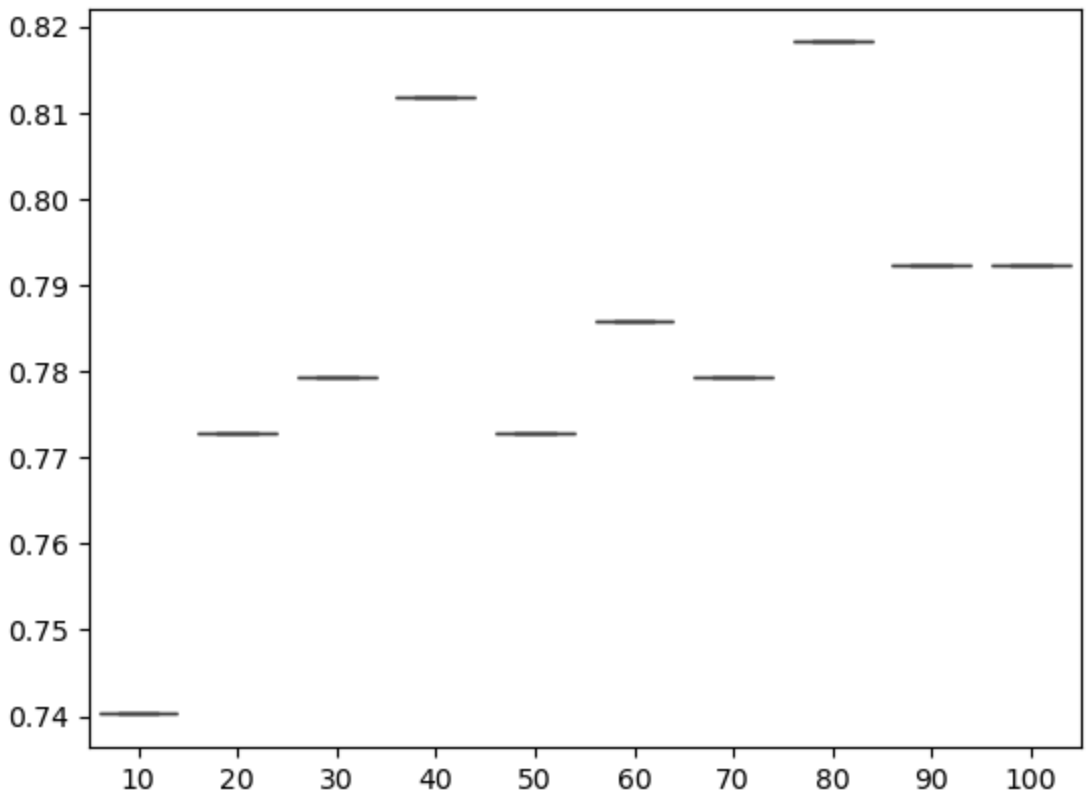
In [86]:
```
score_df.T
```

Out[86]:

	10	20	30	40	50	60	70	80	90	10
score	0.74026	0.772727	0.779221	0.811688	0.772727	0.785714	0.779221	0.818182	0.792208	0.7922(

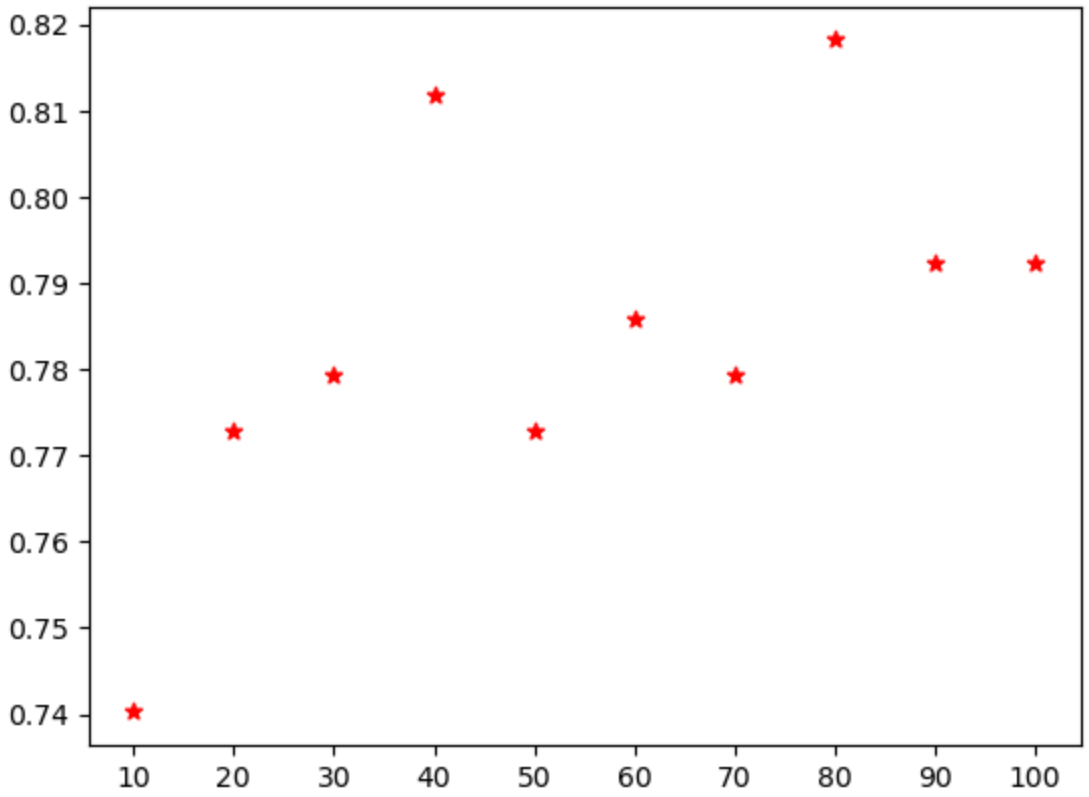
In [87]:
```
sns.boxplot(score_df.T)
```

Out[87]:
```
<Axes: >
```



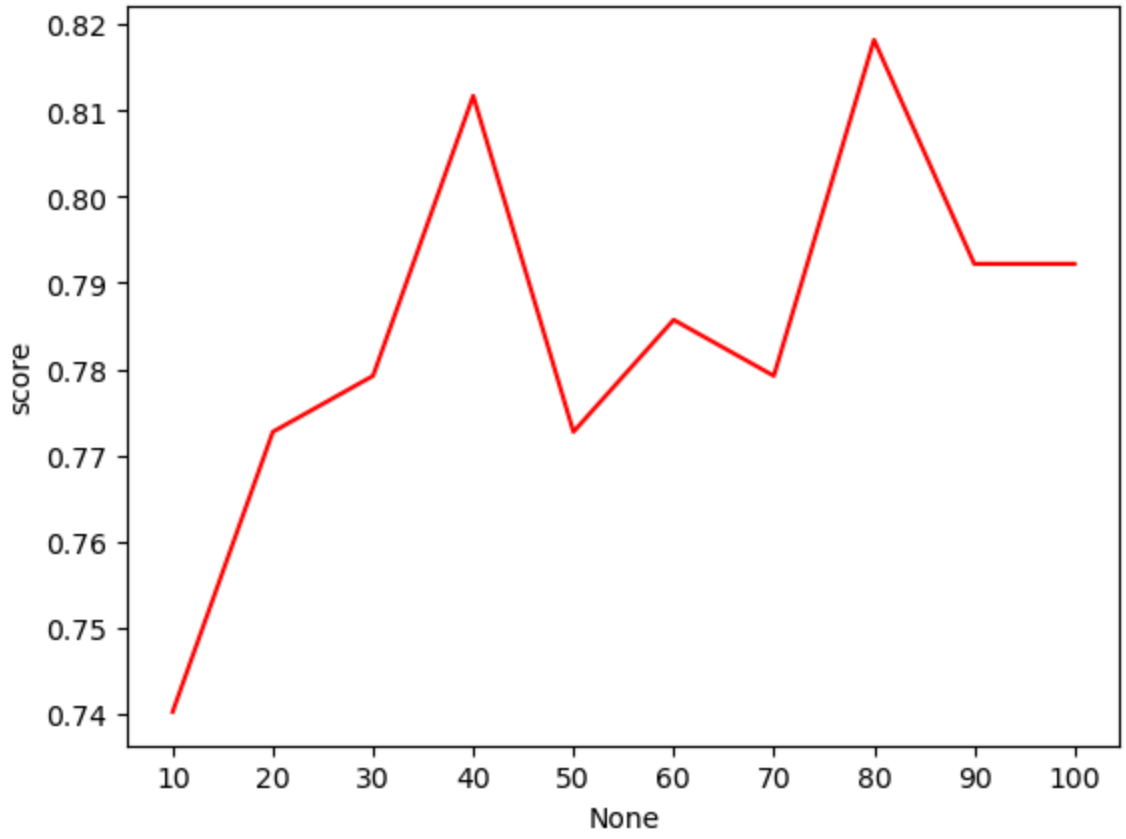
In [88]:
```
plt.scatter(x=score_df.index,y=score_df["score"],c="red",marker="*")
```

Out[88]:
```
<matplotlib.collections.PathCollection at 0x1f31b63e050>
```



In [90]:
```
sns.lineplot(x=score_df.index,y=score_df["score"],c="red")
```

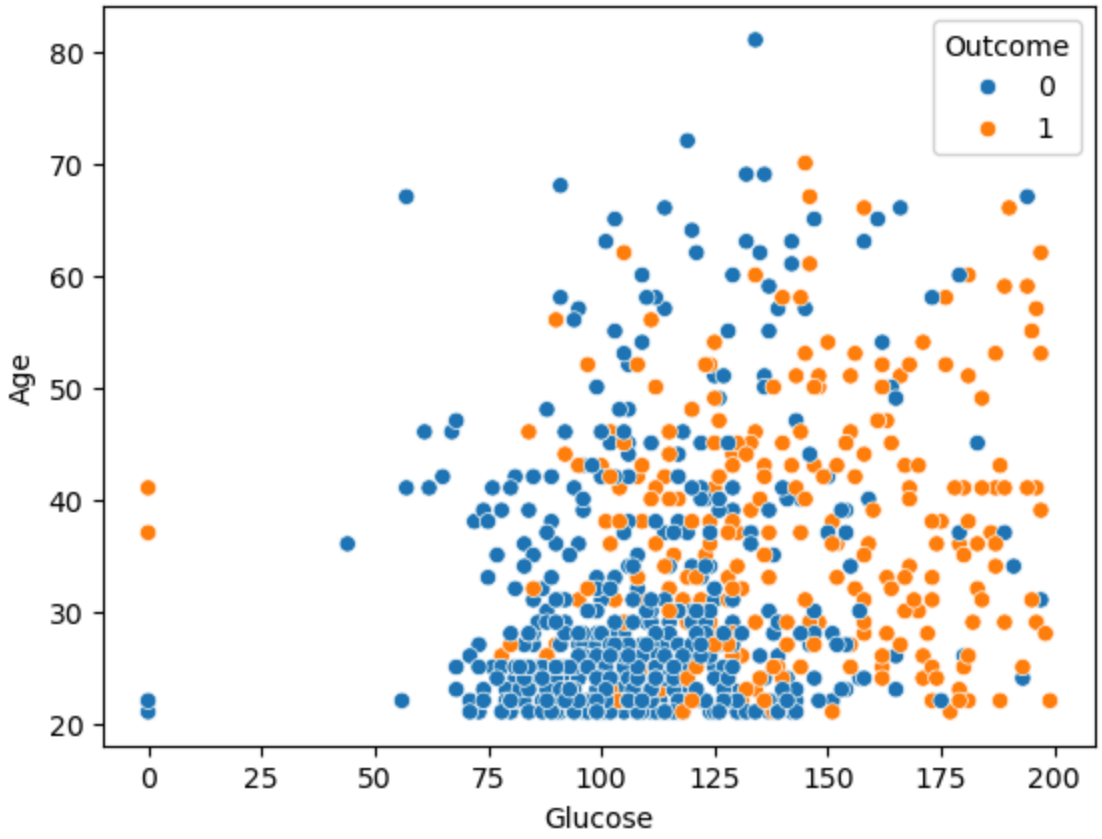
Out[90]:
```
<Axes: xlabel='None', ylabel='score'>
```



In [ ]:
```
# tune decsionTree on max_depth
```

In [94]:
```
sns.scatterplot(x=data["Glucose"],y=data["Age"],hue=data["Outcome"])
```

Out[94]:
```
<Axes: xlabel='Glucose', ylabel='Age'>
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: